Break New Ground

San Francisco
September 16–19, 2019

Running Oracle Database and Applications in Docker Containers on Windows

Christian Shay

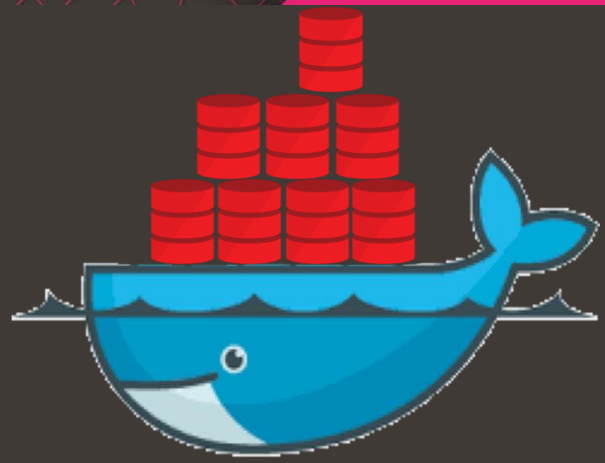
Product Manager
Oracle Database on Windows

Agenda

- Introduction to Containers and Docker
- Oracle Database on Windows Containers
- Instant Client on Windows Containers
- Oracle Data Provider for .NET Core on Nano Containers

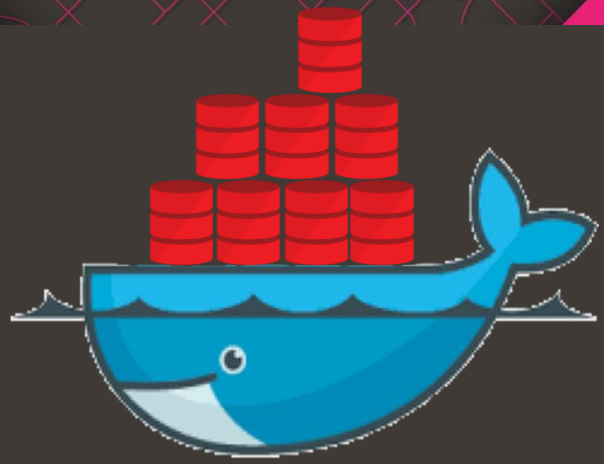
Introduction to Containers and Docker

What are Containers?



- Unlike a VM which provides hardware virtualization, a container provides operating-system-level virtualization by abstracting the “user space”
  - Containers *share* the host system’s kernel with other containers.
- Containers are lightweight (when compared to VMs)
  - No need to install Guest OS
  - Less CPU, RAM, Storage required
  - Can start up and shut down very quickly
- Provide uniformity despite differences between development and deployment

What is Docker?



- A software container platform designed for developing, shipping and running apps leveraging container tech
- Originated from Linux / Linux Containers
- Now also available on Windows and Mac OS X

Docker Terminology

- docker-engine: The host software running the containers
- Images: Collection of software to be run as a container
- Containers: A container on the host OS
- Registry: Place to store and download images
- Volumes: Place to persist data outside the container

Windows Containers: Hyper-V vs Process Level Isolation

- Process level isolation
  - Containers share the same kernel with the host, as well as each other. This is approximately the same as how containers run on Linux.
- Hyper-V isolation
  - Each container runs inside of a special virtual machine. This provides kernel level isolation between each container as well as the container host.

Docker on Windows Requirements

- Windows Server 2016 or later
  - Build 14393
  - Hyper V or Native Windows Containers
- Windows 10 :
  - Windows 10 64bit: Pro, Enterprise or Education (1607 Anniversary Update, Build 14393 or later).
  - Virtualization enabled in BIOS
  - Hyper-V enabled

Installing Docker on Windows Server

- OneGet provider PowerShell module DockerMicrosoftProvider
  - Published by Microsoft, requires elevated Powershell
  - Install-Module -Name DockerMsftProvider -Repository PSGallery –Force
  - Install-Package -Name docker -ProviderName DockerMsftProvider
  - Restart-Computer –Force
- This module also enables Windows Containers in the OS
- More details:
  - https://bit.ly/2m2YPEx

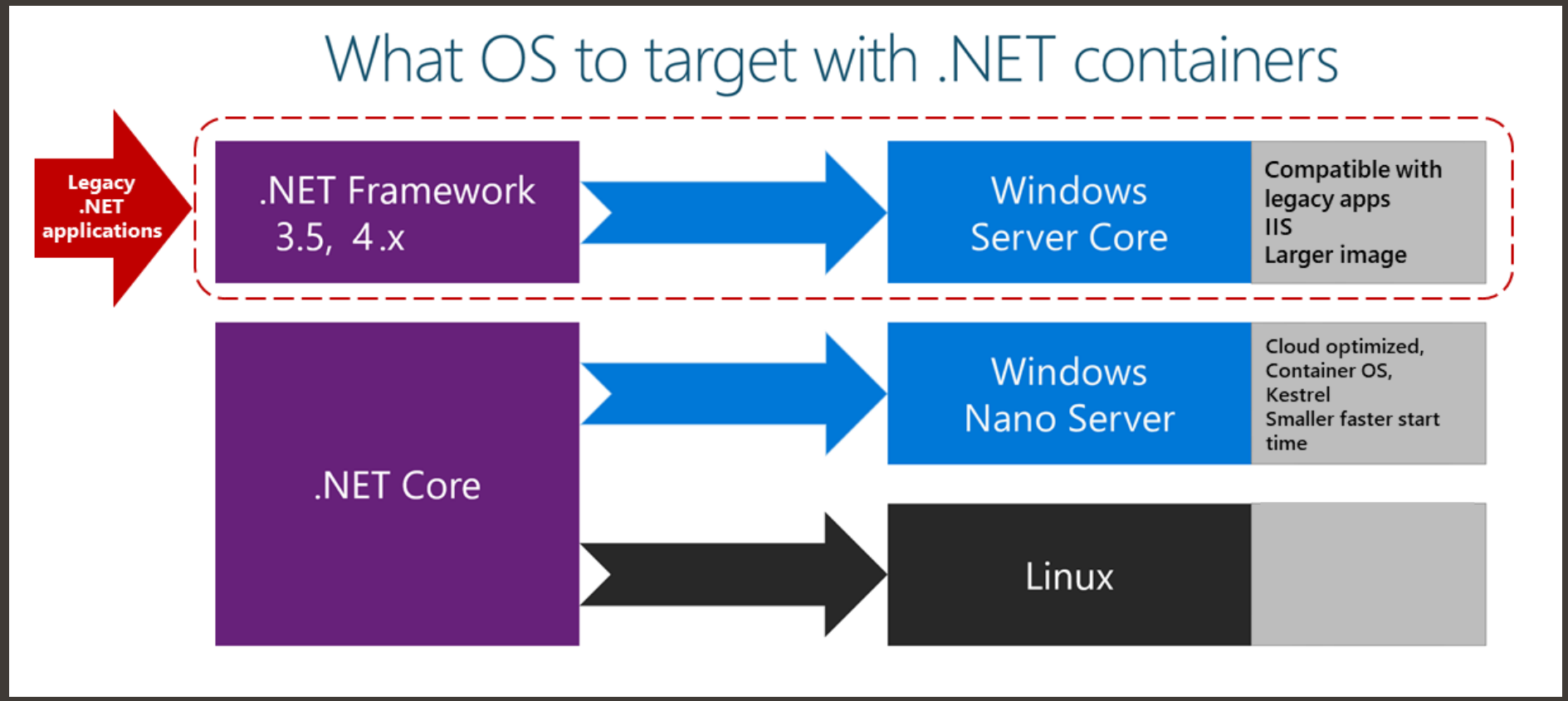
Installing Docker on Windows 10

- Download and Install Docker CE Desktop
  - https://store.docker.com/editions/community/docker-ce-desktop-windows
- After installing, switch from Linux (default) to Windows containers in menu

Images: Server Core, Nano Server, Windows

- Server Core
  - Minimalistic but includes more functionality for legacy apps (eg IIS web server)
  - Many legacy windows apps can run on it (eg Oracle Database, Oracle Client)
  - Moderate size (1.5GB compressed)
- Nano Server
  - Small and fast (100MB compressed)
  - Mostly for .NET Core applications only
  - Other executables may be built using Nano API
  - Missing many features – but MS has added important ones (Limited Powershell support)
- Windows
  - (Mostly) Full featured Windows
  - Gigantic (8+ GB uncompressed)

Server Core vs Nano Server - .NET Support



Docker on Windows Base Images

- Server Core, Nano, and Windows:
  - https://hub.docker.com/r/microsoft/windowsservercore/
  - https://hub.docker.com/r/microsoft/nanoserver/
  - https://hub.docker.com/_/microsoft-windows
- .NET Development
  - https://hub.docker.com/r/microsoft/dotnet/

Container version v Host version compatibilty

- Tight dependencies between version of host and container versions:
  - If using process isolation containers, host should be same version as container
  - If using Hyper-V containers, host should generally be newer than or same as container
  - Compatibility matrix: https://bit.ly/2JSTo5A
- Microsoft has disallowed “latest” tag due to this
  - FROM microsoft/windowsservercore:1709
  - FROM microsoft/nanoserver:1709_KB4043961
  - FROM microsoft/windowsservercore:1803
  - FROM microsoft/dotnet:2.1-aspnetcore-runtime-nanoserver-1803

Oracle Database on Windows Containers

Oracle Database on Windows Containers: How to get it working today

- Do a silent install of Oracle Database using software only install
  - Provide a response file
- Workaround required to avoid “Null pointer exception” issue:
  - Set INVENTORY_LOCATION=C:\Program Files\Oracle\Inventory in the response file
  - Bug 28747182
- Create Database
  - DBCA cannot currently run in docker containers due to bug 28747089
  - So, copy data files, or
  - Generate database create script SQL scripts with DBCA outside of container
  - Scripts have hard coded paths so make sure local drive/directories match container

Oracle Database on Windows Containers: Additional tips

- DBCA 19 no longer puts SYS/SYSTEM/PDBADMIN passwords in scripts. You may need to hard code.
- Use Virtual Account for Oracle Home User
- Allow time for Oracle Database Service to come up before running additional scripts to start listener or database
  - Eg: ping –n 121 127.0.0.1 > nul
- If issues arise, run container in interactive mode and view DBCA script output or silent install logs

Oracle Database on Windows Containers: Demo

- Windows Server 2016 running in Oracle Cloud (Compute)
- Oracle 19.3 installation package for Windows as zip file
- Response file as db.rsp
  - With INVENTORY_LOCATION fix added
- vsredist_x64.exe from
  - http://download.microsoft.com/download/0/5/6/056DCDA9-D667-4E27-8001-8A0C6971D6B1/vcredist_x64.exe
- Database Creation Assistant (DBCA) scripts or database files

Oracle Database on Windows Containers

- Windows Server 2016 running in Oracle Cloud (Compute)
- Uses Server Core base image in dockerfile:
  - FROM mcr.microsoft.com/windows/servercore:ltsc2016

Connect to the database from other containers

- Find out IP address of database container
  - From interactive shell: ipconfig
  - From host OS: Docker ps and Docker inspect container_name
- Find out Service name
  - From interactive shell: lsnrctl services
- Set up TNSNAMES.ORA and SQLNET.ORA in client
- Then connect as usual

DEMONSTRATION

Database in Windows Container

Database Demo: Dockerfile

```
# Base image
FROM mcr.microsoft.com/windows/servercore:ltsc2016
#copy golden db image after taking it from OTN
COPY /WINDOWS.X64_193000_db_home.zip c:/data/db_home.zip
# unzip Oracle golden db image and install VS2013 runtime
RUN powershell -command Expand-Archive c:\\data\\db_home.zip -DestinationPath c:\\data\\db_home
RUN  setx path ".;c:\data\db_home\Bin;%path%;"
ENV ORACLE_HOME c:\\data\\db_home
COPY /vcredist_x64.exe c:/vcredist_x64.exe
#Install VS2013 runtime env
RUN powershell.exe -Command \
    $ErrorActionPreference = 'Stop'; \
    Start-Process c:\vcredist_x64.exe -ArgumentList '/install /passive /norestart ' -Wait ; \
    Remove-Item c:\vcredist_x64.exe –Force
#Provide response file to install Oracle Software Only (modified to workaround bug)
COPY db.rsp c:\\data\\db.rsp
#Execute response file to install Oracle Software Only
RUN c:\data\db_home\setup.bat -silent -responseFile c:\data\db.rsp
```


Database Demo: Dockerfile (part 2)

```
# setup the environment
ENV ORACLE_HOME c:\\data\\db_home
ENV ORACLE_SID orclcdb
ENV CLASSPATH c:\\data\\db_home\\jlib;c:\\data\\db_home\\rdbms\\jlib;
#Expose ports
EXPOSE 5500 1521
#copy dbca generated scripts to create db (may be necessary to hardcode passwords)
COPY /scripts C:/data/app/opc/admin/orclcdb/scripts
#execute scripts to create db
RUN c:/data/app/opc/admin/orclcdb/scripts/orclcdb.bat
#copy post installation scripts
COPY post_install.bat c:/data/scripts/post_install.bat
COPY startdb1.sql c:/data/scripts/startdb1.sql
COPY startdb2.sql c:/data/scripts/startdb2.sql
# Default command to start db
CMD ["c:/data/scripts/post_install.bat"]
```

Database Demo: post_install.bat

- `ping -n 121 127.0.0.1 > nul`
- `sqlplus /nolog @c:\data\scripts\startdb1.sql`
- `orapwd file=PWDorclcdb.ora password=changeoninstall11## entries=100`
- `lsnrctl start`
- `ping -n 61 127.0.0.1 > nul`
- `sqlplus /nolog @c:\data\scripts\startdb2.sql`
- `ipconfig`
- `ping -t localhost > nul`

Database Demo: startdb1.sql

- `connect sys/mypass as sysdba;`
- `alter pluggable database orclpdb open;`
- `exit;`

Database Demo: startdb2.sql

- `connect sys/mypass@localhost/orclpdb as sysdba;`
- `create user hr identified by hr;`
- `grant connect, resource, unlimited tablespace to hr;`
- `exit;`

Demo commands

- docker build -t dboow c:\dbdemo2019
- docker run –d dboow
  - In background
- docker run –it dboow
  - Interactive shell – remove last line in dockerfile

Instant Client on Windows Containers

Oracle Instant Client on Windows Container

- Use Server Core
  - Won't run on Nano Server containers
- Copy instant client zips (from OTN) and extract them
- Set path to point to Instant Client
- That's it!

Demo: Node.js and Oracle Instant Client

- Banana Farmer demo from Chris Jones' blog -- converted to Windows:
  - *A node-oracledb Web Service in Docker*
  - https://bit.ly/2NMFaB6
- Creates a web service that responds to GET, POST, PUT
  - eg GET http://172.17.0.3:3000/bananas/Gita
  - [{"SHIPMENT":"{ \"farmer\": \"Gita\", \"ripeness\": \"All Green\", \"kilograms\": 100 }"}]

Banana Farmer Demo Schema

- CREATE TABLE bananas (shipment VARCHAR2(4000) CHECK (shipment IS JSON));

- INSERT INTO bananas VALUES (
- '{ "farmer": "Gita", "ripeness": "All Green", ”kilograms": 100 }');
- INSERT INTO bananas VALUES (
- '{ "farmer": "Ravi", "ripeness": "Full Yellow", ”kilograms": 90 }');
- INSERT INTO bananas VALUES (
- '{ "farmer": "Mindy", "ripeness": "More Yellow than Green", ”kilograms": 92 }');

Node.js/Instant Client Demo dockerfile

- Copies Instant Client zip and required MSVCRT120.DLL dependency
- Automatically downloads node installation zip file
- Expands zip files
- Copies package.json and Server.js (from Chris Jones' blog)
- Npm install
- installs node-oracledb dependency listed in packages.json
- Set path to include instantclient and node.js
- Start node

DEMONSTRATION

Instant Client in Windows Container

Node.js Demo: Dockerfile

```
# Base image
FROM mcr.microsoft.com/windows/servercore:ltsc2016 as builder
SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue';"]
ENV NODE_VERSION 8.12.0
ENV IC_FILENAME instantclient-basic-windows.x64-18.3.0.0.0dbru.zip
ENV IC_FILENAME2 instantclient-sqlplus-windows.x64-18.3.0.0.0dbru.zip
ENV IC_FOLDER instantclient_18_3
RUN Invoke-WebRequest $('https://nodejs.org/dist/v{0}/node-v{0}-win-x64.zip' -f $env:NODE_VERSION) -OutFile 'node.zip' -UseBasicParsing ; \
    Expand-Archive node.zip -DestinationPath C:\ ; \
    Rename-Item -Path $('C:\node-v{0}-win-x64' -f $env:NODE_VERSION) -NewName 'C:\nodejs';
COPY $IC_FILENAME instantclient.zip
COPY $IC_FILENAME2 instantclient2.zip
COPY msvcr120.dll c:/windows/system32/msvcr120.dll
RUN Expand-Archive instantclient.zip -DestinationPath C:\ ; \
    Expand-Archive instantclient2.zip -DestinationPath C:\ ; \
    Rename-Item -Path $($env:IC_FOLDER -f $env:NODE_VERSION) -NewName 'C:\instantclient';
```

Node.js Demo: Dockerfile (part 2)

```
RUN $env:PATH = 'C:\instantclient;C:\nodejs;{0}' -f $env:PATH ; \
 [Environment]::SetEnvironmentVariable('PATH', $env:PATH, [EnvironmentVariableTarget]::Machine)
RUN mkdir c:/demo
WORKDIR c:/demo
COPY package.json package.json
COPY server.js server.js
RUN npm install
FROM mcr.microsoft.com/windows/servercore:ltsc2016
SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue';"]
COPY --from=builder /nodejs /nodejs
COPY --from=builder /instantclient /instantclient
COPY --from=builder /demo /demo
COPY --from=builder /windows/system32/msvcr120.dll /nodejs/msvcr120.dll
WORKDIR c:/demo
ARG SETX=/M
RUN setx /M PATH $('C:\instantclient;C:\nodejs;'+ $Env:PATH)
CMD ["c:/nodejs/npm.cmd", "start"]
```


Oracle Data Provider for .NET Core on Nano Containers

ODP.NET Core

- Available on nuget.org
  - https://www.nuget.org/packages/Oracle.ManagedDataAccess.Core/
- Supports Windows, Oracle Linux and Red Hat Linux

Oracle Data Provider for .NET Core on Nano Server Demo

- Build .NET Core Web application in Visual Studio or command line
- “Publish” to folder
- Zip up folder
- Move zip into deployment directory

Oracle Data Provider for .NET Core on Nano Server Demo

- Pre-Build ODP.NET app, published to publish.zip
- In dockerfile:
  - FROM mcr.microsoft.com/dotnet/core/runtime:2.1-nanoserver-sac2016
  - (add command to copy and extract your zip file here)
  - COPY PUBLISH /app
  - ENTRYPOINT ["dotnet", "/app/oowdemo.dll"]
- docker run dotnetdemo

DEMONSTRATION

ODP.NET Core in Nano Server

Questions?

Safe Harbor

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at http://www.oracle.com/investor. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.